Food and Nutrition Service

Electronic Benefits Transfer (EBT) Systems Testing Guidelines for the Special Supplemental Nutrition Program for Women, Infants, and Children (WIC)

Final

June 2002

USDA

Booz Allen Hamilton

TABLE OF CONTENTS

1. INTRODUCTION 1

1.1 BACKGROUND 1
1.2 PURPOSE 2
1.3 SCOPE 2
1.4 DOCUMENT OVERVIEW 2

2.0 WIC EBT OVERVIEW 4

2.1 MODEL ARCHITECTURE 4
2.2 TESTING WIC EBT COMPONENTS 6

3.0 TESTING WIC EBT SYSTEMS 9

3.1 TEST OBJECTIVES 9
3.2 COMMON MYTHS OF TESTING 11
3.3 PRINCIPLES 12
3.3.1 Test the Entire System 13
3.3.2 Test the Boundaries of the System 13
3.3.3 Test System Independent of the Developer 13
3.3.4 Test the Likely and Unlikely Scenarios 14
3.3.5 Stop System Development During Testing 14
3.3.6 Start Tests Without Residual Data from Other Tests 15
3.3.7 Simulate the Live Environment as Much as Possible 15
3.3.8 Document Properly and Thoroughly 15
3.3.9 Retest When Necessary 15
3.3.10 Follow Up and Verify Resolution of All Issues and Problems 16
3.4 TEST PHASES 16
3.5 TEST TYPES 17

4.0 TEST PREPARATION 20

4.1 GOALS AND OBJECTIVES 20
4.2 STAFF ROLES AND RESPONSIBILITIES 21
4.3 TEST SCHEDULE AND DURATION 22
4.4 TEST SETUP AND CONFIGURATION REQUIREMENTS 25
4.5 APPROVAL CRITERIA 26

5.0 TEST DOCUMENTATION 28

5.1 REQUIREMENTS TRACEABILITY MATRIX 28
5.2 TEST PLAN 30
5.2.1 Introduction 30
5.2.2 Test Organization 30
5.2.3 Test Environment 30
5.2.4 Test Scripts 31
5.2.5 Test Reporting and Tools 31
5.3 TEST SCRIPTS 32
5.3.1 Benefit Package Information 33
5.3.2 WIC UPC Information 34
5.3.3 Case Information 34

5.3.4 Test Scenarios ... *34*
5.3.5 Testing Multi-program States ... *35*
5.4 TEST PROBLEM REPORT ... 36
5.5 TESTING ISSUES LOG ... 38
5.6 EBT WIC REPORTS ... 39

6.0 TEST EXECUTION ... 41

6.1 ENVIRONMENT OVERVIEW ... 41
6.2 TEST COORDINATION ... 41
6.3 SCRIPT EXECUTION ... 43
6.4 PROBLEM MANAGEMENT ... 43
6.5 RESULTS VALIDATION ... 43
6.6 TEST REPORTING ... 44
6.7 REPORT REVIEW ... 44

7.0 SUMMARY ... 46

APPENDIX A – GLOSSARY OF TERMS ... 47

APPENDIX B – ACRONYMS AND ABBREVIATIONS ... 49

APPENDIX C – TESTING CHECKLIST ... 50

APPENDIX D – SAMPLE TPR FORM ... 51

APPENDIX E – REQUIREMENTS TRACEABILITY MATRIX (RTM) ... 52

APPENDIX F – SAMPLE TEST SCRIPT TEMPLATE ... 53

APPENDIX G – SAMPLE WHAT-IF TESTING SCENARIO FORM ... 58

APPENDIX H – TESTING ISSUES LOG ... 59

APPENDIX I – REFERENCES ... 60

TABLE OF FIGURES AND TABLES

Figure 1. WIC EBT System ... 4
Table 4-1. Test Participant Responsibilities ... 21
Table 4-2. Sample Test Schedule ... 23
Table 5-1. WIC System Functions ... 32
Table 5-2. Defect Rating Scale ... 36

1. INTRODUCTION

System testing is one of the most critical aspects of a system's development effort. Its primary purpose is to determine whether there are defects in the system's operation. It is therefore vital that any errors in the system software and hardware be identified, prioritized, and resolved as early in the testing process as possible. Undetected defects can cause system components to malfunction and can inhibit a successful deployment during both pilot and statewide rollout. In addition, design changes forced by software flaws discovered after implementation are more costly than repairing anomalies identified during testing.

More than identifying design and software problems, system testing determines whether the development effort has met the technical design specifications. A comprehensive program of system testing offers the critical opportunity to observe system performance under both normal and abnormal conditions. Abnormal conditions are set up to exceed normal system operational environments, thereby ensuring dependability.

This document discusses the necessity to develop sound and reliable test plans and procedures for the Electronic Benefits Transfer (EBT) system of the Special Supplemental Nutrition Program for Women, Infants, and Children (WIC), established under the U.S. Department of Agriculture (USDA). It begins with an overview of two varieties of WIC EBT systems, and then discusses the various aspects of system testing, test preparation, and test documentation. Examples of best-known methods and lessons learned are provided.

1.1 BACKGROUND

The WIC program was established under Section 17 of the Child Nutrition Act of 1966 (Public Law [P.L.] 89-642, 80 Stat. 885, Oct. 11, 1966). The purpose of the act is as follows:

> In recognition of the demonstrated relationship between food and good nutrition and the capacity of children to develop and learn, based on the years of cumulative successful experience under the national school lunch program with its significant contributions in the field of applied nutrition research, it is hereby declared to be the policy of Congress that these efforts shall be extended, expanded, and strengthened under the authority of the Secretary of Agriculture as a measure to safeguard the health and well-being of the Nation's children, and to encourage the domestic consumption of agricultural and other goods, by assisting States, through grants-in-aid and other means, to meet more effectively the nutritional needs of our children.[1]

[1] *Child Nutrition Act of 1966, Public Law 89-642, page 2-2.*

Section 17 of the authorizing legislation instructed the Secretary of Agriculture to establish a program to provide food assistance and nutritional education to at-risk pregnant, postpartum, and breastfeeding women, as well as infants and young children. A component of this program is EBT.

EBT is the conversion of paper-based food instruments (i.e., checks or vouchers) to electronic food instruments (i.e., smart cards or magnetic stripe cards). Electronic food instrument transactions rely on commercial point-of-sale (POS) networks and a state-managed settlement system that produces automated clearinghouse (ACH) files for initiating grocer payment by each state's clearinghouse. The total dollar value of transactions due to settle the next business day is drawn down nightly from each state's respective letter of credit, maintained by the Federal Government. These funds are then delivered to the participating retailers' bank accounts thus reimbursing for the EBT transactions.

1.2 Purpose

This document is designed as a guide to assist the states as they plan, prepare for, and execute testing of their WIC EBT systems. To provide insight into testing, the document discusses best-known industry methods as well as lessons learned. Since each state must take ownership of its EBT system, this document will equip the states with pertinent information and guidance to uncover hidden system defects before the system is rolled out.

1.3 Scope

In providing tools for planning and conducting the various tests of the WIC EBT system, this document includes principles, methodologies, and areas of testing. Guidance for planning, documenting, and executing successful system testing is also provided. This document will not include a comprehensive set of test scripts for testing a WIC EBT system. The state is encouraged to contact FNS for a copy of actual test scripts from various WIC EBT states. However, a generic test script template is provided. The document's intended audience is states preparing to implement and test a WIC EBT system, as well as states in the early stages of testing systems under development.

1.4 Document Overview

The guidelines provided herein are designed to assist state WIC agency personnel in preparing for and executing WIC EBT system tests. The document is not meant to serve as a test plan but rather as a guide to assist state personnel in preparing their individual system test plans. The document contains seven separate sections as follows:

- Section 1: Introduction – discusses the background, purpose, and scope of this document.

- Section 2: WIC EBT Overview – provides an introduction to system testing and an overview of WIC EBT.

- Section 3: Testing the WIC EBT System – defines the objectives of system testing, outlines a number of testing principles, and discusses the various phases and types of system testing.

- Section 4: Test Preparation – provides a detailed description of the planning required to conduct successful system testing.

- Section 5: Test Documentation – discusses a variety of system test tools, including the test plan, test scripts, and the test problem reporting process.

- Section 6: Test Execution – steps through a typical WIC EBT system.

- Section 7: Summary – provides a document summary and best practices.

- Appendices – includes nine separate appendices, including a glossary of terms, a sample test script, and a number of other useful resources.

2.0 WIC EBT OVERVIEW

As discussed in Section 1, the purpose of testing is to ensure that each component and interface between components, and the system as a whole, functions as designed. Each state has adopted its own approach in implementing WIC EBT, which has resulted in different designs and test scripts. However, even though the designs are different, each system has commonality of components and functions. This section presents the general architecture of a WIC EBT system and provides guidance for testing the common areas of the system.

2.1 MODEL ARCHITECTURE

The WIC EBT system involves the following participants: local and state WIC agencies, recipient, financial intermediary, retailer, EBT processor, manufacturer, USDA Food and Nutrition Service (FNS), and Federal Reserve Bank. Figure 1 illustrates the relationships and data flow between the different entities of the WIC EBT architecture.

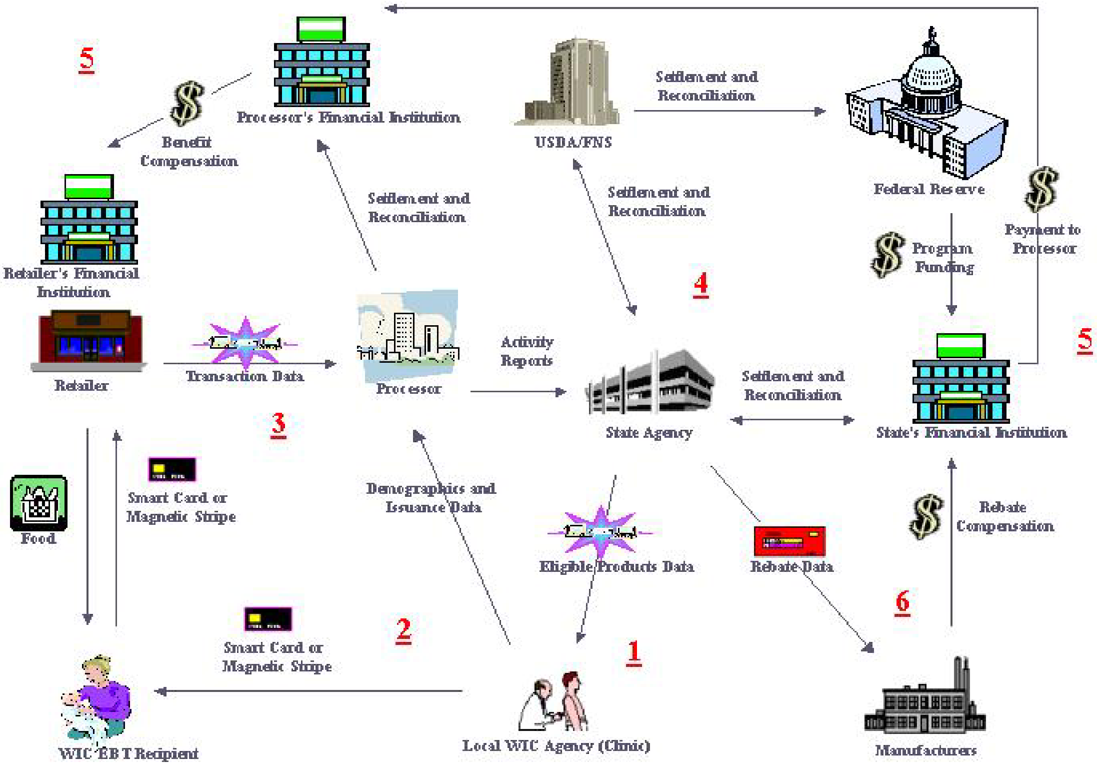
5
Processor's Financial Institution
Benefit Compensation
USDA/FNS
Settlement and Reconciliation
Federal Reserve
Retailer's Financial Institution
Settlement and Reconciliation
Settlement and Reconciliation
Payment to Processor
Program Funding
4
5
Retailer
Transaction Data
Processor
Activity Reports
State Agency
Settlement and Reconciliation
State's Financial Institution
3
Smart Card or Magnetic Stripe
Food
Demographics and Issuance Data
Eligible Products Data
Rebate Data
Rebate Compensation
6
2
Smart Card or Magnetic Stripe
1
WIC EBT Recipient
Local WIC Agency (Clinic)
Manufacturers

Figure 1. WIC EBT System

Figure 1 encapsulates the complete cycle of benefit eligibility, issuance, distribution, redemption, and payment to the retailer. The process begins at the Local WIC Agency (clinic) where the recipient applies for the benefits. The steps of the WIC EBT process, outlined in Figure 1, are described below:

Step 1: A Certified Professional Authority determines eligibility for WIC assistance at WIC clinics based on nutritional risk and income. Eligible candidates receive nutritional assistance in the form of a prescribed food package. The prescribed food package contains food items appropriate for the particular individual's nutritional needs. The clinics send the food package information and demographic data to the processor and establish the participant account.

Step 2: Participants are issued a payment initiation device, typically a card. The card may be a smart card[2] in which case the benefits data will either be on the card when it is taken from the clinic or it will be transferred to the card when it is first used at one of the specialized readers installed at a retailer or another convenient location designated by the WIC State agency. The card may also be a magnetic stripe card, which will function in a manner unique to the WIC program. Currently there are no operational on-line WIC EBT systems that use the commercial retail electronic payment infrastructure, but such an application is not outside the realm of possibility and should not be precluded. (The Michigan WIC Program is currently planning a magnetic stripe card solution that will utilize the commercial infrastructure for retailer settlement.) The clinic may give the participant a personal identification number (PIN) when the card is first issued or provide an option to select the PIN at some other convenient time. A PIN must be established prior to first use of the card regardless of the type of card issued.

Step 3: Recipients use the WIC EBT cards at authorized retailers to purchase WIC-specific food items prescribed by the local WIC clinic. Electronic WIC transaction data (purchases, returns, voids, reversals, account balance inquiries) are transmitted daily to the EBT processor, which in turn provides activity reports (defined in the request for proposal) to the state agency for settlement. An EBT processor-generated settlement sheet gives the aggregate balances calculated at the end of the business day. The EBT processor then creates an ACH file that is sent to a concentrator bank, instructing that the net total of all purchases be subtracted from a clearing account and delivered to the third-party processor and retailers that provided the acquired transactions and food to WIC recipients.

Step 4: The state agency ensures that the settlement sheet is balanced, then directs its financial institution or state treasury to fund the clearing account used by the processor for paying merchants and third-party processors.

Step 5: Once federal program funds are received, the state's financial institution forwards the funds to the processor's financial institution.

Step 6: The state agency administering the WIC program uses transaction reconciliation information regularly to prepare rebate data. The rebate data is then forwarded to the food

[2] A smart card is a plastic card with the same physical dimensions as a credit card but contains a microprocessor (in WIC applications), random access memory, and International ISO compliant mechanical contact terminals for interactions with card readers.

manufacturers, who issue rebate payments to the WIC state agency. This program income is subsequently used to fund future WIC benefits. (The rebate system is not an automated process.)

Testing of the WIC EBT system typically includes the following areas: local clinic, EBT card and processor, retailer, financial settlement and reconciliation, and call center. EBT operates quite differently from its predecessor paper-based system.

With EBT, the entire process of benefits distribution, redemption, and reporting is automated. Specific differentiations are further described in Section 2.2.

There is also a significant difference between the smart card and the magnetic stripe card used for benefit distribution. These differences have a significant impact on testing. They are discussed in Section 2.2.

2.2 Testing WIC EBT Components

Major WIC EBT components include the EBT card (which provides the necessary authentication for accessing benefits) and the local clinic, EBT processor, and retailer point-of-sale (POS) terminal systems. The local clinic and the retailer POS terminal system interface with the EBT card and the EBT processor system. The local clinic system provides recipient account setup and issues benefits. The EBT processor receives and maintains demographic and benefit issuance information from the local clinic system, authorizes redemption claims submitted by the retailer, and produces daily and monthly financial reports of benefit activity. The retailer system facilitates the exchange of food items with the benefits conveyed on the EBT card. For the entire system to work, each component must communicate effectively as well as process data consistently.

The following major functions of the components and their interfaces should be tested thoroughly to ensure accurate and timely delivery of benefits to the participant:

Local Clinic System

- Account Setup
- Benefit Issuance
- Card Issuance
- Card Replacement
- Conversion to Category and Subcategory
- Access Daily Transaction History
- System User Accounts and Privileges
- Interface with the Card
- Interface with the EBT Processor

EBT Processor System

- Interface with the Local Clinic System

- Account Setup
- Benefit Issuance
- Online Account Setup and Benefit Issuance
- Card Issuance
- Card Replacement (This may include test of the process to replace hybrid cards for WIC/FS and/or other program partnerships. For example, there will have to be an interface with the FSP EBT system.)
- PIN Management
- UPC Database Maintenance
- Card and Account Status Maintenance
- "Hot" Card List Maintenance
- Daily Transaction Processing – Transaction Authorization
- Transaction Validation
- Daily and Monthly Reports Generation
- Rebate Tracking
- Settlement and Funds Movement
- Call Center Support

Retailer System

- Card Holder Authentication (e.g. PIN)
- Card Validation
- WIC Approved Item Validation
- Period of Benefit Access Validation
- Insufficient Benefits on the Card
- Interface with the EBT Processor
- POS Management Functions (User, Manager Ids and Passwords)
- Backroom PC (if applicable)
- Third-party processors (TPPs). (It will be only a matter of time before a WIC State encounters a situation when a retailer opts to use a TPP for the WIC transactions. In those States that will be certifying the individual store systems, then an acceptance test will have to comprehend this system configuration.)

Smart Card

- Benefits Storage and Retrieval
- Benefits Debit and Credit
- Card Holder Authentication
- Transaction History Log
- Card/Program Lock and Unlock
- PIN Selection and Change

Magnetic Stripe

- Primary Account Number Storage

- Required Number of Tracks
- Required Track 2 Format
- Required Coercivity
- Required Color Scheme
- Required Security Features

As can be seen above, the role and capabilities of smart cards are more significant than those of the magnetic stripe cards in a WIC EBT system. Based on the design of an off-line smart card WIC EBT system, specific tests should be conducted around card issuance, card replacement, balance transfer from old card to new, benefits expiration at the end of each month, maintenance of statused card list, and locking and unlocking of different programs on the card.

System tests must also be developed to address the differences between "standalone" and "integrated" systems. It is important to note that it is no longer true that only standalone systems will be deployed by WIC State agencies. For example, both Texas and New Mexico will be developing integrated systems, and the States must certify these integrated systems before they can process WIC EBT transactions. There are differences in the testing of the two systems. For example, a single scan per food item on the integrated systems versus two scans per food item on the standalone systems.

System tests may include testing interfaces with the FSP EBT system or other interfaces with potential program partners.

It is also strongly recommended that a system security test be performed, along with concurrent processing and system capacity testing. Security testing involves areas within the WIC EBT system that could be exploited, either intentionally or unintentionally, to the detriment of any EBT system component: for example, protect the system from unauthorized use of the clinic system, benefits and card issuance, and addition of benefits to the card. Concurrent processing is the initiation of the same operation or function, from the same type of device, at the same time (e.g., performing two POS debit transactions on two POS devices simultaneously). This test ensures that proper record lock mechanisms are implemented and that data corruption is avoided. Capacity testing simulates high-volume transaction data within the system to ensure that it can sufficiently support caseload volume anticipated in the production.

Specific test scenarios are dependent on system design, but the testing areas indicated above remain common to the different WIC EBT implementations. Further guidance for preparing and executing test scenarios is provided in the remaining sections of this document.

3.0 TESTING WIC EBT SYSTEMS

Section 2 identified components of both the on-line and off-line WIC EBT system and indicated areas to test. This and subsequent sections address the preparation and execution of a test.

Despite different types of tests and that testing takes place in all phases of system development, every test must begin with clear objectives. This section introduces the concept of testing beginning with a review of test objectives, principles, types, and phases. Subsequent sections will build on these concepts to help the state prepare and execute a test.

3.1 Test Objectives

A good system test includes clear objectives and goals. This section describes general test objectives, including how they are created and used in a system test, and provides examples of each objective.

- **Objective 1: Validate functionality and behavior according to the design and requirements**

  The first testing objective is verification that the system functions according to user requirements and detail design. Test scripts should be crafted to validate that the system meets requirements and is consistent with the detail design documentation. A critical review of user requirements and detail design is essential to determine whether there are mutually exclusive requirements or if the design itself does not address all requirements. Overall, the test plan and scripts should start from simple to complex using the system design as the basic document. The functionality of each component can be tested separately with additional scripts of increase complexity but related to the functions, to replicate operational activity. The elements of design that are implicit can be drawn up as specific tests. What-if's scripts can be used to supplement the test by adding scenarios such as negative testing (entering alpha characters to numeric-only fields, generating error messages to demonstrate error handling and/or security edits).

- **Objective 2: Ensure that the system correctly handles normal conditions**

  As indicated in Section 2, WIC EBT systems contain numerous components and subcomponents that routinely transmit data across the system, update system databases, log activity, settle accounts, generate reports, and monitor system use and growth. The test team must test drive all routine functions, observe the results, and record deviations from expected behavior as well as all defects.

Observations of deviations must be noted and communicated to the developer for possible modification. Defects must be logged and described completely so that others will be able to re-create the defect.

- **Objective 3: Ensure that the system correctly handles error conditions**

  Although a system may handle typical transactions exceptionally well and the tester may presume normal functionality testing on the part of the developer, the bulk of software contained in a WIC EBT system is outside the "normal" flow of traffic. Therefore, after testing the system for compliance with what it was designed to do, the system must be tested for its ability to handle unexpected events. The software that detects and handles errors is typically invisible in most documentation, but it is the area of the software that contains the most problems. This is because the number of errors that can enter the transactions stream has untold combinations, any one of which could have disastrous consequences. Clear differences exist between proper and faulty error handling, but it is sometimes difficult to discern whether a detected error response is a defect or rather the proper handling of an error condition. For example if communication networks fail it is expected the system will detect the failure, alert operators, and ensure that all messages in the various queues throughout the system are correctly handled. It is also expected that some messages will be lost because they were traveling through the communication line when it was disrupted. This loss of message is not a defect. However if no component exists to ensure proper accounting for the lost message or adjust the accounts affected by the missing message, then a defect clearly exists.

- **Objective 4: Uncover system defects.**

  This objective is not so much an objective as it is a result of the objectives listed earlier in this section. Certainly if no defects exist then the system test has met its objectives. However, "uncover system defects" is listed separately to reinforce the importance of testers using their range of skills and creativity to develop effective scenarios for uncovering defects. It is not enough to test only the normal paths and expected errors the system was designed to detect and handle. Retail payment systems such as WIC EBT move requests and responses continuously, and each transaction contains unique data that will never be repeated over the entire life of the system. This fact alone ensures that no test will ever cover the myriad outcomes for all transactions; testers are thus forced to generalize about the data contained in the transactions. Typically, defects will be discovered when messages pass from one subsystem to another, for example from a smart card to a reader or the transmittal of a nightly batch from the clinic or retailer POS systems. If the messaging is interrupted, redial sequences must reset the transmittal or discard partial information. Depending on telecom processes, different testing and error handling conditions may be involved. Some of the movements can occur within micro- or milliseconds; testers must therefore use their creativity to discover messaging defects during message transition and other likely areas for defective logic. Scenarios need to be created that may not actually occur in the system but that will nevertheless test the tester's hypothesis for an event that could occur in the system. For example, it is not likely that an interface cable to a smart card reader will become unplugged at a critical point in a transaction. However, a disruption in power can occur at any time through a sudden loss of power in a store.

- **Objective 5: Verify file format standards.**

  This objective will verify that messages and file formats are developed and utilized per established standards as documented in X9.93 2002 – Financial Transaction Messages – Electronic Benefits Transfer. This objective is primarily handled in the design phase but is verified in unit testing.

3.2 Common Myths Of Testing

There exist some common myths associated with system testing that are worth discussing because they may incorrectly shape a tester's approach and prevent achievement of the objectives discussed earlier. False assumptions prevent establishing solid testing procedures and the discovery of defects. Some of the common testing myths are listed and discussed below:

- **Myth:** *If changes are made to only one of the components, you do not have to test the entire system*. In fact, when a system component is modified, a ripple effect may affect other system components. When new versions of software or hardware are released and incorporated into the system, complete regression testing should be considered. This kind of testing can ensure compatibility of new components with existing ones. Care must be taken in identifying other components, functions, or modules. Regression testing should be performed to ensure that a fix made in one place did not affect the system in another.

- **Myth:** *If a state is replacing its current system or a component thereof with a system or component previously in operation elsewhere, or if a state is importing an operating system as its first-time system, end-to-end testing is not needed.* In fact, in such a scenario each system must be reviewed and completely tested, independently. Further, if any system component is different, the entire integrated system again must be tested independently. It is dangerous to assume that a component or core system that worked for one state will function the same way for another. Situations have occurred where a complete, end-to-end test was not conducted because the developer convinced both the state and FNS that various components of the new system were unchanged from the previous operating version. A state must ignore such claims and conduct a complete test.

- **Myth:** *Unusual scenarios should not be tested because it is unlikely they will occur in a real environment*. Although unusual scenarios by nature do not occur frequently, they nevertheless can and do occur and may cause serious system and financial damage. They must be tested, and any system problems must be resolved.

- **Myth:** *Testing is a cumbersome process that should be avoided*. In fact, tests are necessary to prevent catastrophic failures in production.

- **Myth:** *System testing should be the sole responsibility of the developers*. Although system developers have the most knowledge and insight regarding the system, an inherent conflict of interest exists among developers who perform testing. This is because developers are overly familiar with the system and also tend to see it more on the basis of how it works rather than on how it might fail.

- **Myth:** *States are not responsible for User Acceptance Testing*. In fact, states must run their own user acceptance test scripts. They should not take the developer's word that the system works properly and therefore does not need to be tested. States also should not rely solely on outside sources, such as consultants, to run the state's user acceptance test scripts. The scripts should be run before outside sources run their test scripts and what-if scenarios.

Unfortunately many other testing myths exist. Test participants need to understand the test scope and requirements and stay focused on achieving the test objectives.

3.3 PRINCIPLES

Testing objectives are goals; testing principles are guideposts to those goals, allowing unbiased appraisals. Principles provide structure to a test and prevent misunderstandings, thereby helping all test participants to find consensus in the test results. The following principles outlined and discussed below will afford a greater possibility of success in testing, allowing all test participants to know what to expect during the test.

1. **Test the entire system.**

2. **Test the boundaries of the system.**

3. **Test system independent of the developer.**

4. **Test the likely and unlikely scenarios.**

5. **Stop system development during testing.**

6. **Start tests without residual data from other tests.**

7. **Simulate the live environment as much as possible.**

8. **Document properly and thoroughly.**

9. Retest when necessary.

10. Follow up and verify resolution of all issues and problems.

3.3.1 Test the Entire System

Components other than core components have a direct effect on the core components themselves. Unless the entire system is tested end-to-end, a tester cannot attain the requisite level of confidence that a system is functional. When a particular component is modified, the tendency is to test the component and stop testing; however even in this scenario the entire system must be tested for potential ripple effects and compatibility issues resulting from the change. All system components are linked by interfaces and dependencies; changes to one component may affect other components elsewhere in the system.

3.3.2 Test the Boundaries of the System

When testing a system, the tester must ask the following questions:

- What are the acceptable input values?
- What are the acceptable menu selections?
- What are the acceptable configurations?
- What is the predictable outcome?
- What is the actual outcome?

Testing a system with specific input values does not satisfy all possible conditions the system could experience. It is not enough that the system works for certain allowable input values; the state must ensure that the system accepts all allowable input values and rejects all unallowable ones. System design typically provides specific guidance in identifying the boundaries of a function. For example, consider a system that specifies values ranging from 1 to 10 as allowed input values. The system should be tested not only with values 1 through 10 but also values 0 and 11, which are not valid.

3.3.3 Test System Independent of the Developer

The system developer is all too familiar with the system and its behavior. The developer may therefore be aware of existing issues with the software or hardware and may employ workarounds during testing, presenting a conflict of interest. The motivation of the developer is to attain system approval from the client to receive compensation for services rendered. Therefore, the developer may not be motivated to find faults in the system because such faults would result in additional time and resources in fixing them. Although the developer may assist in the setup and coordination of tests, a party independent of or different from the developer should lead the testing. The tester must also ensure that the test setup does not remove conditions that would otherwise be present in an operational system.

The state must be confident that the system works according to requirements: The testing phase is an opportunity for the state to gain familiarity with the system and to determine whether the system behaves as envisioned. An additional independent party may be enlisted by the state to assist in testing. For the formal Federal acceptance test, the independent parties will be FNS staff and staff from our consultants. If a bug is found, a problem report is written for the developer, explaining the test run and the error returned. The developer can make a fix, but it’s up to the testers to determine when to incorporate the fix and continue with testing.

3.3.4 Test the Likely and Unlikely Scenarios

System testing must incorporate scenarios that would occur during both normal and abnormal system use. Varying knowledge and skill levels among the testers can establish a diverse perspective of the testing environment. A diverse testing team not only provides a skilled technical capability but also a more fundamental and comprehensive analysis of the design requirements from a user’s point of view.

3.3.5 Stop System Development During Testing

Freeze the code! Consider the possible ripple effects of any system modification during testing and stop further system development. Code freezing should occur after completion of Unit and Integration testing. Code modifications are inevitable in some circumstances; however, changes during the final phase of testing require a thorough and adequate regression test. During end-to-end testing, version releases on hardware and software will complicate and potentially jeopardize progress.

The system must be tested after all software development has ceased. As discussed earlier, a change in system code may bring adverse or unexpected effects to the remainder of the system. For example, consider an actual incident that occurred during off-line system testing: The test participants allowed the developer to correct a minor problem of the system after the problem was discovered during testing. The fix was seemingly harmless; however after it was made, a functionality that had been verified only a few steps before was now malfunctioning.

An important lesson learned from this experience is that when a change, however small, is made, the whole system may be affected. Thus (as per principle 1 above) the entire system must be tested.

3.3.6 Start Tests Without Residual Data from Other Tests

All EBT systems have a financial component. The systems must track all funds entering and leaving the system, as well as track funds (transactions) remaining in the system. A WIC EBT system is more complex because it tracks benefits by food items, rebates, and funds. If accounts are increased or depleted before testing begins, ensuring the accuracy of the reports becomes an increasingly difficult chore. If the reports shows double the benefits the tester expected, then the tester must either track down the source of issuance or write the anomaly to the test log as a potential defect. If the original issuance data was destroyed, then the test must begin from the start, at a loss of valuable time and resources. The same holds true for all financial components of a WIC system. If the test shows that rebates are running at triple the expected amount because of a long period of prior testing, it will be virtually impossible to show what benefits were redeemed or whether the rebate report is correct. As a consideration, the state should print reports prior to the start of a test to reflect the system balance, account, card, and UPC database status, and any outstanding transactions.

3.3.7 Simulate the Live Environment as Much as Possible

The test system must represent all possible configurations of the live environment. For example, if there will be stand-alone and multi-lane POS terminal devices in the live environment; they must be represented in the test system configuration. It would not be enough to test only the stand-alone POS terminal setup. Testing should also include all clinic system configurations and communication links.

3.3.8 Document Properly and Thoroughly

When problems arise in system functionality or behavior, describe the symptoms and not the tester’s suspicions. This method is necessary for effective problem resolution. Provide as much relevant information as possible, including test step number, test station, test component, card number, and account number.

3.3.9 Retest When Necessary

Know when to stop testing. Establishing thresholds will help determine whether continued testing will prove beneficial or whether the system needs to be reverted back to development. Too many major errors during a system test are an indication that the system is unstable and not ready for testing. The system should be retested later, after thorough testing by the developer.

3.3.10 Follow Up and Verify Resolution of All Issues and Problems

System functionalities can be verified at each test step. Some functionality, such as settlement and reconciliation, can be performed only after the developer has generated the reports. These functionalities may also need to be matched up with POS terminal receipts. Ensure that all results are available and verified prior to approving the system. Do not approve a system until all problems and issues are resolved. The developer may verbally agree to resolve issues and problems, but the state must follow up with the developer until all issues are resolved. Unless the developer is under contract for a maintenance phase, once the system is approved for rollout, the developer is not obligated to address any further issues and problems.

3.4 Test Phases

This section describes in detail the various stages of testing. The discussion addresses the purpose of each phase, scope of the test, and who typically conducts the test. The following WIC EBT relevant test phases will be addressed in detail:

- **Unit Testing** – The developer must test each component or subcomponent before it is integrated into the system. This is known as unit testing and is the developer's responsibility. Although the state is not involved at this testing stage, it can and should perform its own unit testing to compare its results to those of the developer. This is also the stage of testing which ensures that edit rules are followed for each field, such as valid date and name formats.

- **Integration Testing** – After components have passed unit testing, they can be configured and appropriately linked for testing of the integrated system. This is known as integration testing and is conducted by the developer. The focus of this testing phase is to ensure that the interfaces and functionality are correct and work together between related components but not as part of the system as a whole. Compatibility among subsystems is a key area that must be tested in this phase.

- **Interface Testing** – Interface testing conducted by the developer, involves testing the data exchange between two or more components after they are installed on the system. Once the components are on the system, tests can be run for generating reports.

- **Functional Testing** – As the name suggests, functional testing is designed to demonstrate that the system functions according to specifications. It is typically conducted between the state and the processor, and it should test all components and functions of the system. Prior to functional testing, a test script should be developed that thoroughly tests all functions of the system. Testing results should be documented, and all open issues should be resolved prior to the next phase of testing (acceptance testing). Open issues refer to test results that differ from expected results, or to items discovered during testing that the state would like changed or presented in a different way than originally specified. Issues affecting system processing should be resolved prior to acceptance testing, but those issues deemed cosmetic or minor may remain open for acceptance testing.

- **User (Sign-off) Acceptance Testing (UAT)** – User acceptance testing is a method by which a system is validated through a series of well-documented tests encompassing all functional areas of the system. This testing must be conducted exclusively by the state agency prior to federal acceptance testing, with the purpose of "wringing out" the system. Testing results must be documented, and all problems must be identified and prioritized by their severity and impact on the system. Any problem related changes affecting system processing should be resolved prior to the federal acceptance test. User acceptance testing may include regression testing. Beta or "field testing" can also be included within the acceptance testing process. (Regression and beta/field testing are discussed further in Section 3.5.) Federal acceptance testing will not occur until after the state has successfully concluded its acceptance testing.

  Note: It is paramount to freeze code changes during this and subsequent testing phases.

- **Federal Acceptance Testing** – Each state will have its own federal acceptance test after the state UAT has been completed, documented, and affirmed a success. Each state is responsible for overseeing the federal acceptance test and for conducting a complete, end-to-end test. Situations have occurred where a complete end-to-end test was not conducted because the developer had convinced the state and FNS that no components of the new system had been changed from the previous operating version. A state must ignore such claims and conduct a complete test. Federal acceptance testing typically serves as the formal justification for federal approval that allows a state to move forward toward system pilot and eventually to system rollout.

- **Live Test** – Just prior to pilot implementation in each state, a pre-Go live field test is planned to test the environment with a limited group of test participants. This test serves as a final verification, albeit on a limited scale, that the EBT system is functioning end-to-end within the actual operating environment. The results of the Live Test could prevent the system from "going live," even if FNS has approved the Federal Acceptance Test.

3.5 Test Types

Within the various phases of testing, specific types of tests should be conducted depending on the nature of the logic or function being tested. The following types of testing can be conducted at various phases of testing:

- **Boundary Testing** – As discussed above, boundary testing involves testing the limits of acceptable input values. For example, if a system component accepts values 1 to 10, then entering values 0, 1, 10, and 11 would be considered as testing the boundaries.

- **Security Testing** – Security is defined as any area within the EBT system that could be exploited, intentionally or unintentionally, to the detriment of any EBT system component.

- **Concurrent Processing Testing** – Concurrent processing testing is defined as initiation of the same operation or function, from the same type of device, simultaneously (e.g., performing two POS debit transaction on two POS devices simultaneously). Testing this area involves causing components to compete for system resources simultaneously.

- **Performance Testing** – Performance testing ensures the system meets the transaction response times set forth in the system requirements. This is particularly of interest to the retailers because delays at the lane are prohibitive in their business process. It also impacts the call center and clinic staff performance if timely data is unavailable to support recipients or retailers. Thus, during testing the state should ensure the transaction times are within the specified performance requirements.

- **Stress Testing** – Stress testing examines the EBT system's ability to maintain its integrity under peak loads during system operation. This type of testing should be discussed in advance with the developer because setup can be very involved. Stress Testing is designed to simulate near-production demand on the system to determine whether the system can handle the load of all processes that will run concurrently in production. The main goal in stress testing is to identify those jobs that are not being dispatched efficiently and to make the necessary changes to ensure the system is able to process all jobs efficiently within the capacity of the software and hardware. Stress test results may also help determine the frequency at which certain jobs are performed. It also provides the opportunity to fine tune any locking rules that may be in place as well as escalations on various tables prior to the EBT system Go-live.

- **What-if Testing** – What-if testing examines the system's response in unusual and unlikely scenarios. Test scripts must be prepared in detail to test specific scenarios. The tester should not be concerned about whether a scenario is too unlikely or complex; it is better to test a what-if scenario and resolve all doubts than to only assume the system will function properly. What-if testing should include an expected result and the actual results received from the test. These results can be compared to ensure that the results returned are what the user was expecting to see according to the specifications.

- **Ad Hoc Testing** – Ad hoc testing is performed after the completion of the scripted testing. Ad hoc testing involves testers running random tests to try to "break the system." This type of testing is usually not documented unless a bug is discovered or the results returned are not what the tester was expecting. Examples of ad hoc testing include, but not limited to invalid date formats, alphabetic characters used in digit-only fields, required fields missing in the transaction, and invalid field lengths. Note that ad-hoc testing may cause data corruption; therefore, it is advisable to perform this test at the conclusion of the formal testing.

- **Regression Testing** – Regression testing is designed to validate at some subsequent time the functionality that was problematic during original testing. However, the test should not be limited to this scope. If code modification has occurred after the original test, the testing principles (Section 3.3) recommend that all system functionality be thoroughly tested during a regression test. Risk is involved in testing only the functions that produced problems; at a minimum all components that interface with the problematic component should be retested to ensure the new code did not affect the system adversely.

- **Contingency Testing** – Contingency testing determines whether a system can provide an acceptable level of service during an emergency. For example, if the primary processor were not operational, then the process of switching to a backup processor and its capabilities would be tested.

- **Beta Testing** – In software development, a beta test is the second phase of software testing, where a sampling of the intended audience tests the product. Originally the term *alpha test* meant the first phase of testing in the software development process. The first phase includes unit testing, component testing, and system testing. Beta testing can be considered pre-release testing, when the software is available for selected users to conduct normal operating conditions in the kind of environment in which the software will be used. Beta testing is performed during user acceptance testing.

4.0 TEST PREPARATION

This section focuses on the planning and logistical aspects of organizing, executing, and documenting a successful systems test. Thorough planning helps ensure that a test meets its objectives, is fully executed within the project deadline, and is completed within its allotted budget. It is also a federal requirement to develop a comprehensive test plan and execute a successful acceptance test.

4.1 Goals and Objectives

Before the first test script is executed, it is critical that the goals and objectives of the test be established. Without defining pass-fail criteria and an acceptable tolerance range for defects, the state will not be able to determine whether the system provides all required functionality and whether the provided functionality operates properly.

The overall goal of systems testing is to ensure connectivity of all components, integration of all subsystems, and emulation of a full processing cycle, all without any major defects or discrepancies. A discrepancy is defined as an outcome other than the expected result of a prepared test script. All discrepancies must be systematically and meticulously recorded to achieve successful resolution. System test scripts should thoroughly test and map the objectives of the major functional areas of the system to the expected outcomes, based on the specifications and requirements as defined in the system requirements document.

Planning and management of a system test is an extensive but important phase. The test plan must have been thoroughly developed, involving the coordination, scheduling, and effort of all test participants. Project milestones must be integrated within the overall project management timelines. A thorough test plan should be organized and structured to avoid the critical and costly error of a second test.

The system test environment should emulate an actual production (live) WIC environment (e.g., single- and multi-lane configurations). It is recommended that the test take place on location at the state and not at the developer's site, which provides the developer the opportunity to demonstrate that the system can function properly outside the laboratory environment. Testing at a state site can also lead to additional tests of the system, including testing of telecommunication functions within the state's environment, which may be different than the developer's laboratory conditions.

The interface among all functional components of the system should be tested to replicate a life cycle WIC process. A full description of the test environment is essential and should be documented to include exact configurations for personal computer (PC) terminals, POS devices, and the telecommunication network for each functional component of the system to assist in evaluating test results. The hardware and network specifications, as well as the system design, can be referenced in the system detail design document. Any disparity with respect to the test configuration should be documented, with ramifications noted.

4.2 Staff Roles and Responsibilities

This section discusses the importance of assigning a test manager, and of receiving individual commitments for participation in the system test. Testing can be a time-consuming process that requires dedicated resources for extended periods of time. Testers should represent a cross-section of all potential users to ensure that the system meets the requirements for all user types. Table 4-1 lists the responsibilities of test participants, the recommended number of participants (in parentheses next to indicated participant), and additional comments.

Table 4-1. Test Participant Responsibilities

Test Participants	Responsibilities	Comments
Test Manager (1)	• Oversees and manages all aspects of the test • Organizes meetings, staffing, and assigns responsibilities for test scripts • Leads daily discussions on issues and problems encountered during the test	Ensure activities are compliant with scripts and that any deviations are properly documented. Maintain a master copy of the test scripts. Manage time and closely track card inventory to prevent loss. Role does not necessarily have to be assigned to the EBT processor or vendor.
Technical Support Team (2-3)	• Provides ad hoc support in system configuration • Troubleshoot problems • Provides support in setting up special cases	The support team prepares the test configuration, establishes communication links, and verifies proper software version prior to the actual testing.
Local WIC Office/County (1–2)	• Gathers eligibility information • Administers benefits • Sets up card cases • Ensures clinic system is functional and accurate	Document findings and issues relevant to the operation of the WIC office. Communicate finding to the state and the vendor.
State Representatives/ Health and Human Services (2-3)	• Maintains databases • Interfaces with EBT system	Send benefits authorizations. Set up accounts.
FNS Representatives (1)	• Reviews project documents • Observes test(s) • Provides supplemental and what-if test scripts • Participates in testing	Recommend approval or disapproval of the system after the test.

Test Participants	Responsibilities	Comments
Third-Party Merchants (1-2)	- Integration testing	Ensure retailer system is operable.
EBT Processor (2-3)	- Provides system test plan and scripts - Supports all phases of test - Provides assistance and documentation necessary to validate test results - Provides technical support	Scripts and scenarios should encompass all combinations of applicable transactions, including exceptional conditions (e.g., non sufficient funds [NSF] transactions). A minimum of three transactions per type is recommended to generate sufficient traffic and to fully evaluate the system.
IV & V Contractor (2-3) *(Optional)* Note: If the IV&V Contractor is intended to be the FNS EBT consultants, then these people will be mandatory, not optional to the test.	- Reviews test plan - Provides what-if scenarios - Participates in testing - Recommends approval or disapproval of the system	Test all areas of the system and document all findings. Prepare reports to FNS with recommendation for approval or disapproval of the system.
Customer Service (1-2)	- Reviews the system from the call center perspective	The call center staff representative familiarizes with the system functionality and user interface
Other Program Partners (1-2)	- Test their respective program for proper functionality	Applies to systems that support multiple state programs

4.3 TEST SCHEDULE AND DURATION

For a system test to be successful, time management is crucial. Establishing start and end dates are critical to the test proceeding in a timely manner. A clear definition of the test duration should assist in acquiring commitments from various resources. To ensure that test participants fully understand their roles and responsibilities, a comprehensive test schedule should be developed, to include information about test location, setup and delivery of testing equipment at each location, and the daily schedule of testing events.

Testing should be simulated to allow crossover from one month to the next to ensure accuracy of benefit issuance and redemption and to validate monthly processing functions, (e.g., day 2 of the test is the last day of the month and day 3 is the first day of the next month). It is recommended that the system clock not be changed to simulate this crossover on the same day, as this would interfere with the normal system process and might introduce other unanticipated problems. The ideal is to allow the system to transition from one month to the next in real time, which thus would test the system under natural conditions.

Depending on the complexity of the WIC EBT system and the number of test scenarios, testing may require 3 to 5 days. Based on past experiences with WIC EBT systems, scheduling a test for less than 3 days is not recommended. It is better to perform a thorough test before rollout than to possibly have to make modifications afterwards. Equipment setup and preparation, what-if testing, and settlement and reconciliation review should also be incorporated into the test schedule. It is often best to start with simpler transactions during the early days of testing to ensure that functionalities exist, then perform the more complex sequential tests, and then save the what-if scenarios for the latter part of the test. This method makes it easier to examine reports with simpler data transactions and reporting. The following sample test schedule presents recommendations and activities for a full 5-day systems test:

Table 4-2. Sample Test Schedule

Test Day	Test Activities
Day 1	• Logging all cards used for the test including the conditions and information stored on each card • Categorize test scripts for each test day • Generate host reports • UPC and food category table maintenance • First-time settlement and user creation at the clinic system • Card setup functionality at the WIC clinic system • Uploading a list of statused cards to the system • Card authentication • Terminal authentication • WIC clinic system on-line card setup and benefit issuance • Retailer setup and first-time initialization • Cashier and manager file maintenance at the POS • WIC purchases • WIC POS in-lane reasonability checks • WIC purchase exception testing • WIC clinic system network tests • WIC clinic system user security tests and other functionality • Card security test on a POS device • Call Center functionality test • Supplemental issuances for current month and issuances for next month • Settlement of all systems • What-if testing
Day 2 (Last day of the month)	• Generate host reports: daily for Day 1 • WIC card set up for initial benefits to be received the following month • WIC on-line instrument adjustment • WIC next-month instrument adjustment (staged) • Posting staged WIC benefits and transactions • Exception testing: Duplicate issuances • Recipient number change and card replacement • All WIC POS transactions • Cross-test between programs using hybrid cards

Test Day	Test Activities
Day 2 cont'd	• Modification of host UPC table • PIN and stale date locking of cards • POS/customer service restoration • Changing issuance sites and re-staging benefits • Creating and staging card locks at customer service • Supplemental issuances for the next month • Settlement of all systems • What-if testing
Day 3 (First day of next month)	• Generate host reports: daily for Day 2 and monthly/weekly for Month 1 • Applying card and program locks to cards at the POS • Unlocking cards at the WIC clinic system • Posting WIC recurring and initial benefits • Posting WIC benefits adjustments • Posting benefits re-staged to new issuance sites • Posting staged transactions from Day 2, including refunds, purchase reversals, and manual transactions • Manual and staged POS transactions • Other POS tests, such as cashier and manager file functions and shutdown • WIC clinic system on-line card setup and benefit issuance • WIC clinic system off-line card setup and benefit issuance • Several card replacement scenarios • Creating and staging card locks at customer service • Supplemental issuances for current month; incorrect recipient IDs to test benefit recall on Day 4 • Settlement of all systems • What-if testing
Day 4	• Generate host reports: daily for Day 3 • Apply card and program locks to cards at the POS • Unlock cards at the WIC clinic system • Post WIC initial benefits staged the previous day • Post restoral transactions • Post staged POS transactions from Day 3 • Post transfer of value for cards replaced on Day 2 • Retailer deactivation/activation • Settle Soon/Now and Settle Override tests at POS • WIC clinic system verify-only functions, such as checking batch and audit trail backups and performing read-only card operations • Send file containing corrected recipient IDs and accompanying supplemental benefit issuances • Settlement of all systems • What-if testing
Day 5	• Generate host reports: daily for Day 4 • Post WIC transfer of value and accompanying transactions • Post benefit effective Day 5 • Noting activated/deactivated retailers in issuance site listing at WIC clinic systems • Run batch cycle to generate host reports: daily for Day 5 and

Test Day	Test Activities
	monthly/weekly for Month 2 • Exception test • Stress testing • What-if testing

Note: Each day of testing should include the following: 1) a download of the hot card list to the stores, 2) an attempted use of a statused card, 3) UPC and food category table maintenance, 4) Call Center functionality.

There should be more than one day of cross-testing between programs using hybrid cards. This segment of the testing will have to be more fully developed as there are multiple activities that must be tested such a card issuance, card statusing, and Call Center responsibilities associated with hybrid cards.

4.4 Test Setup and Configuration Requirements

Proper test setup and configuration helps to secure an effective and reliable test environment. All relevant system equipment must be available to test all functional areas of the system. The equipment should include hardware and software required to access the system both internally and externally. Testers must also have access to the back-end processes (i.e., stored reports, transaction settlement information, etc.).

The following items should be checked off for testing:

- Most recent production software version loaded
- System parameters set correctly
- Test data and hardware configuration consistent with test plan
- All necessary communication links are available

All system and test documentation required to verify test results must also be readily accessible during the test. A crucial aspect of test setup is preparing sample data for all types of potential transactions. By preparing this data ahead of time, a substantial amount of time can be saved; the test will proceed more quickly. Following are the minimum documentation and equipment that must be prepared prior to the test:

Documentation

- EBT system architecture
- Acceptance test scripts
- Acceptance flow and activities
- System Design Document
- Testing schedule
- Quick reference guide for response and error codes
- Quick reference guide for settlement and reconciliation.

Hardware

- System hardware schematic
- Host configuration
- POS terminals
- Telecommunications network
- WIC clinic system
- Cards and card-related equipment.

Card Management System

- Card issuance (a sufficient number of cards to run the test based on test script scenarios)
- Card replacement
- Retailer selection (simulate two to three stores, single- and multi-lane)
- Card inventory
- Card maintenance.

Reports

- Daily activity reports
- Daily settlement reports
- Monthly reports.

Another area of preparation is ensuring that certain expectations are communicated to the system developer before the test. When a test is progressing according to schedule and all system components are functioning as expected, it is tempting to correct minor problems and bring closure to the testing. The natural tendency is to correct the problem by making code changes, test the specific function, and close the incident report. However there is not enough time to evaluate the effect of the code change on other system functions. Corrections made abruptly usually introduce new problems in the system that could go undetected. Therefore developers should be informed that no changes be made to the system code during a test. It is imperative to make a distinction between code changes and parameter changes however. Parameter changes can be performed to correct minor problems during the test and to bring closure to minor problems, because such changes do not affect the core code of the system. Code changes, as explained above, need to be examined carefully, as they could affect other areas of the system.

4.5 Approval Criteria

This section focuses on determining whether or not a test is successful and whether to proceed to the next phase of development. Test exit criteria are based on test goals and objectives established during the planning process. The number and magnitude of defects play a role in determining whether a system passes or fails a test.

The system test plan should encompass a full range of scenarios to fully test system functionality, while also incorporating exception, stress, and what-if testing. Prior to the start of the test, it is imperative that all test participants fully understand the process and format for documenting and evaluating test results. The following information should be addressed and communicated before the test:

- Defect Reporting Procedure
- Defect Severity Rating System
- Defect Evaluation Criteria

A process should be established whereby all defects uncovered during the test are communicated to the test manager and system processor, and also recorded immediately on a test problem report (TPR) for further review and documentation. (Refer to a sample TPR in Appendix D) Typically the process would include how to document the defect, and when and to whom to report the defect.

A defect rating system must be clearly defined prior to the start of the test. It will assist in determining the severity of a defect uncovered during the test. The rating system should include defined, logical priority levels to assign to each defect. Section 5.4 contains a suggested defect rating method.

At the end of each testing day, sufficient time should be allotted for review of any discrepancies recorded during the day. The entire test team should be provided with the TPR(s) for each defect uncovered during the day and, with total consensus, should assign the appropriate rating/priority level. (Refer to Table 5-2 for an example of a defect rating scale.)

A pass-fail decision on whether to start up a WIC system is based on comprehensive test scripts designed to thoroughly test all functions provided by all components of the system. Pass-fail criteria constitute an implicit understanding among all test participants prior to the start of a test that all defects rated at a certain level (level 4 is recommended) or higher be resolved before the system is approved and placed into production. The developer should commit to correcting level 3 or higher defects by a certain date. On the last day of testing, a final TPR log should be reviewed and updated for accuracy, and a final copy indicating all outstanding defects should be provided. A decision to pass or fail a test should be determined after careful review of all TPRs, documentation, and reports resulting from the test.

5.0 TEST DOCUMENTATION

The previous section focused on test preparations and the environment surrounding the WIC test. Once test preparations are established, the WIC system is ready to be tested. Numerous methods are used to test a system; each approach, format, and sequence of steps will vary according to each state as well as the processor chosen.

This document has discussed numerous approaches to assist testers in assessing the WIC system. This section discusses documentation of the methods used and the results obtained for each test. Documentation tools that can be used during each stage of the process are recommended, and each subsection discusses the purpose and advantages of its respective tool. Examples of these documentation tools are located in the appendices.

This section consists of the following subsections:

- Requirements Traceability Matrix (RTM)
- Test Plan
- Test Scripts
- Test Problem Report (TPR)
- Testing Issues Log
- System-generated Reports

It is strongly recommended that the State, or its system developer, have available at the acceptance test an installed copy of a data mining tool software such as CAST. This is software that has modules in the various programming languages.

5.1 Requirements Traceability Matrix

Before a WIC system can be designed or developed, the system's users, in partnership with the developers, must first understand the system requirements. The Software Requirements Specification (SRS) document specifies what is expected from the system. Because this document forms the foundation of the system, it can be useful in testing the system's functional areas. System requirements and functions of each system component are listed in the SRS document, categorized according to the logical areas of the system. During the design phase of the system project, every requirement is further explained in the Detailed Design Document (DDD) for the system. The DDD is the blueprint used by developers to construct the system.

Once the requirements have been defined and the system has been developed, the system is ready to be tested. At this point the detailed list of requirements is placed in the RTM. The matrix is essential in determining the functionalities of the WIC system, all of which must be tested. The RTM connects each requirement to a specific design element of the system and helps determine whether a project's objectives have been met and user expectations satisfied. The RTM also helps in determining what methods and components are used to meet a project's objectives.

The RTM is composed of rows and columns. The format (number of columns and rows, data to be tracked, numbering scheme, etc.) of each RTM will vary from state to state. However every requirement should be accurately described, down to the lowest possible level. The sample RTM presented in Appendix E contains six columns, five rows, and the following elements. Additional elements (i.e., "Explanation" or "Comments") can be added to the matrix as deemed necessary by each state.

Unique Identification Number
A unique number given to every requirement. This ID number will start with either "UR" or "SR" and will end with a number assigned in ascending order. An identification beginning with UR will be used to reference a user requirement. SR will be used to specify a system requirement.

Requirement
A description of the requirements of the system. Descriptions should be straightforward and easy to interpret.

Requirements document reference number
The paragraph number referencing where the requirement can be found in the Requirements document.

Components
Each requirement can be traced back to a function or component of a system. Components used to satisfy this requirement are displayed in this column.

Test script reference number
The number cross-referencing a test script to the requirement.

Successful/Unsuccessful
A check in the appropriate box indicates whether the requirement tested resulted in success or failure.

Once the system requirements have been agreed on, they can be loaded into an RTM. The RTM can be used throughout various stages of the project's life cycle, from design to implementation. Any changes performed on the system that affect any requirement should be documented in the RTM. The RTM can also be used to manage change and to provide a basis for the test plan and test scripts (explained later in this chapter).

5.2 Test Plan

The previous section discussed the importance of determining requirements for a system and then capturing these requirements in an RTM. Once a system has been designed and developed to the requirements, a system test can be performed to help detect errors in the system and ensure that the system is operating efficiently. Developing an effective system test plan is vital to ensuring a test is performed accurately and proficiently.

A test plan establishes the test's objectives, defines the location and duration of the test, and clearly indicates what activities will be executed during the test. A well-defined test plan provides a clear breakdown of the system's functional groups along with the expected performance of each. The comprehensive test plan comprises the following elements:

- Introduction
- Test organization
- Test environment
- Test scripts
- Test reporting and tools

5.2.1 Introduction

The introduction section of the test plan describes the project and phases that constitute the test. A brief explanation of the test objectives and the tasks required to meet these goals is also provided. In the test objectives area, it is important to state the scope and limitations of the test and to identify the criteria that will determine a successful test. If the location site and duration of the test is already known, this information should be included in the introduction. The introduction to the test plan can also identify the intended audience of the document.

5.2.2 Test Organization

The location of each test and a schedule of activities to be performed each day are provided in the test organization section. Schedule estimates and milestones can help the team perform more efficiently. For those involved in performing test activities, it is important to know exactly what is expected. Roles and responsibilities should be established in the test plan: This will open lines of communication between key players of the test team. Section 4.2 provides a more detailed explanation of each staff member's roles and responsibilities. With only a brief time period to accomplish the test, it is important to coordinate and organize activities to be performed by each group or person involved.

5.2.3 Test Environment

The test environment section of the test plan will explain the significance of a stable testing environment. A detailed description of the test system's architecture as well as each component's functionality should be included. Diagrams and their corresponding explanations should be provided to clarify the system. The WIC EBT system is composed of numerous components: The test plan must specify not only what WIC hardware and software system components will be tested or demonstrated but also what will not be examined. The operating system, telecommunications, connectivity, data configuration, interfaces to other system, and other setup requirements for the test should also be stated in the test environment section. Test setup prerequisites, such as setting up numerous accounts and cards containing different balances, benefit amounts, benefit availability and expiration dates, family sizes, and other elements, should be defined. It is essential that the test environment closely resemble the live environment; if the test environment does not represent all the components of a live system environment, the differences between the two environments, and the reasons for these differences, should be explained.

5.2.4 Test Scripts

Test scripts provide a step-by-step explanation of the actions needing to be performed during a test, categorized by each functional area. Each test script is developed to assess all functions and conditions of the system to be tested. The outcome of each test script is compared with the expected result. This section of the test plan provides a high-level overview of the test scripts to be used during testing, by functional area. Functional areas include: benefit issuance, authorization and redemption, settlement and reconciliation, and administrative functions. The quantity of test scripts is not limited; there should be a sufficient amount of test scripts to thoroughly test each functional area (listed above) before the system goes into production. The focus should therefore be on the quality of the test scripts, not quantity. The purpose of test scripts and their components is further explained in Section 5.3.

5.2.5 Test Reporting and Tools

Each test will vary according to the methods and techniques used in conducting the test. Whether part of the test is automated or conducted manually, it is extremely important to document the events and results of each test while each test activity is being performed. The test reporting and tools section of the test plan lists the tools used to record and interpret all activities performed during the test. Test scripts and test activity logs are only two examples of tools that are further explained in this chapter. Tools also help the test team track any unexpected issues encountered during the test. After a test is conducted, it is important to determine whether the objective of each test scenario was met, resulting in the expected outcome. Various rating scales are used to determine whether a particular function of the system passed or failed the requirements. The defect rating scale is a tool that is used to critique each test result. (An example of a rating scale is provided in

Section 5.4.) If the function tested did not fulfill the conditions and requirements set forth, those standards must be met before the system is released.

Each state's test plan is dependent on its own WIC system. An overall system test plan may consist of multiple test plans, depending on the various stages of testing. Interface, Functional Demonstration, and User Acceptance are but a few examples of testing phases. Numerous test plans may exist depending on the makeup of a state's system and on what aspects and functionalities of the WIC system will be assessed at one time. The state's approach in conducting a test will also determine how a test plan is presented. Test plans also differ according to the layout of each test plan section as well as the content contained in each.

5.3 Test Scripts

A well-executed test contains a comprehensive and well-defined set of test scripts. Test scripts, based on a number of test scenarios, are used to assess the requirements established by system users and to evaluate the functional performance of the system to meet these requirements. Test scripts vary from one test approach to another and can also depend on the test methodology being executed. Since a well-documented test script should contain expected outcomes and actual results, they assess the validity of each system's function and help detect errors in the system. A sample test script is provided in Appendix F, and test script elements are further described in this section. Actual test scripts from WIC EBT states can be obtained from FNS.

Test scripts test the main functions of the WIC system and should include the following areas: benefit issuance, benefit authorization, benefit redemption, settlement and reconciliation, and administrative functions. Each area can be further broken down into individual functions, as shown in Table 5-1.

Table 5-1. WIC System Functions

Test Area	Functions
Benefit Issuance	• Account processing • Card issuance • Card reissuance • Addition and deletion of recipients • PIN issuance
Benefit Authorization	• Card distribution • UPC and food category table maintenance • Hot card list • Client and card inquiry and card update • Expired benefits • Food package data

Test Area	Functions
Benefit Redemption	• Purchases of WIC and non-WIC items • Account balance • Card removal at various stages of the transaction • Unit of measurements • Manual voucher purchases • Credit/Debit WIC • PIN entry • Card status • Retailer functions • Error/reason codes
Settlement and Reconciliation	• Reconciliation of benefits issuance and host reports • Printing and validating terminal receipts • Category/subcategory table updates • Adjustments
Administrative Functions	• Account inquiries and updates • Lock card • PIN change • Transaction history searches • Transfer of funds • Cancel terminal transactions • Call Center functions

Note: One of the functions that must be tested is the interface to the FSP. Maybe that is comprehended in the Call Center functions, but the interface with the FSP will occur at other points as well. The Administrative Test area should contain a requirement to test system reports, other than host reports

A comprehensive test script, which can be created from each of the functional areas listed above, contains the following information (it is important that each test script include this information on the client account being tested):

- Benefit package information
- WIC UPC information
- Case information
- Test scenarios

5.3.1 Benefit Package Information

Each client is assigned to a particular nutritional risk category, and is authorized to purchase a set amount of specific WIC-approved food items. For example a client assigned to the "Pregnant Women" category can be authorized to purchase the following eligible food items: 2 gallons of milk, 2 dozen eggs, 36 ounces of cereal, and 3 pounds of cheese. The type and amount of food items authorized is generally consistent among participants with the same nutritional risks, however, food packages may be tailored to meet unique nutritional needs. Benefit package information provided in the test scripts contains a food package ID (unique to each group), description, category for the grocery item, subcategory, food package description, and quantity.

5.3.2 WIC UPC Information

The WIC UPC database provides information related to every WIC-approved item, including the category identification number, subcategory identification number, UPC number, product description, unit of measurement, and maximum price.

5.3.3 Case Information

Multiple WIC family members belong to one family group; every account is assigned a single Family Group ID Number. Family group information provides details held within every WIC account, including Family Group ID Number, Head of Household member, Representative member or "proxy" (if assigned), card Primary Account Number (PAN), PIN, and card status. Only one primary member in every account is authorized to access the account benefits. A single card is distributed per family group. Demographic information is stored on every WIC card.

5.3.4 Test Scenarios

Test scenarios within each test script can be created from each functional area (displayed above). A test scenario, which is given a unique number, provides the tester with a set of steps to perform when testing each system's functionality. For example when testing the card locking mechanism after invalid PIN attempts, the following steps are to be performed within the test scenario:

1. Swipe Card
2. Enter Invalid PIN (first try)
3. Enter Invalid PIN (second try)
4. Enter Invalid PIN (third try)
5. Enter Invalid (fourth try)
6. Terminal will display "Return to Clinic"

The indicated steps should consist of specific key words to direct the tester in administering a test case. Every test scenario must include a client's beginning and ending balances to reflect what products were affected during the test transaction. The expected result attained from executing each step must also be included. Actual results are compared against what was expected from the system at the time of execution, which allows an assessment of the validity of each system's function.

Test scripts encapsulate all scenarios involved in testing functionality within the WIC system. It is an acceptable practice to test various functions of the system using a single card, just as is it acceptable to use individual cards to test individual functions of the system. The goal is to render the test transactions similar or as close as possible to a live environment. A well-defined test script includes a clear sequence of steps and provides information pertaining to the client's account, before and after the test is executed. Unanticipated results should be recorded and evaluated against the expected results. Section 5.4 discusses the documentation process for evaluating unexpected results.

In addition to test scripts, what-if scenarios may be used during the system test. Note that these scenarios should not impact the remaining test scripts. For example, buying a food item that impacts the balance remaining on the card for a future scenario may undermine the testing process. What-if scenarios, created by test participants, test the limitations of the system. They not only address system functions, they also disclose system weaknesses. All minimum and maximum restrictions of the system are tested. As an example, what-if scenarios can be used to check the system's ability to detect an incorrect header and trailer record format for incoming files. They can also be used to test a client's invalid PIN attempts. A what-if scenario may include the following fields: unique number, a description of the scenario, expected results, actual results, and comments. A sample what-if scenario form is provided in Appendix G.

5.3.5 Testing Multi-program States

States that offer more than one benefit program, such as Food Stamp and WIC, should perform cross testing between the different programs. Some states may choose to provide more than one benefit program on a single card. When testing multi-program states, the following areas need to be considered, and the system must be able to accommodate these functions seamlessly:

Card replacement: A card that has more than one program needs to maintain all its information when it is being replaced. The two or more benefit programs accessible from the original card must be transferred to the new card.

Program locking: The system should enable the user to lock one program on a smart card while keeping the other program active. Even though these programs reside on the same card, they should be independent of one another. This allows the recipient to receive uninterrupted service from one program even though a second program is locked.

Funds co-mingled: Funds between different programs residing on smart card should be kept independent of one another. Each benefit program should have separate accounting.

Recipient information: The system must know what type of information can be shared among the different programs. Information such as recipient name and address can be public (sharable) among the different programs because this information is common across programs. Private data elements, or information that is unique to each program, such as a recipient's nutritional risk, may not be shared.

Call Center: Testing should include a test of the Call Center functionality serving the multiple programs.

Multi-Program Interface Testing

5.4 Test Problem Report

After testing each function of the WIC system and recording the results, it is time to identify all problematic issues of the system. The actual results of each test should be compared with the expected results. This comparison may be accomplished immediately after the test has been performed or at the end of each test day. Problems may arise during the test that may halt the test: These are exceptional conditions, unlike the minor problems discussed in Section 4.4, that if not corrected will prevent the test from proceeding. If these exceptions occur, the test team must stop the test, assess and fix the problem, and conduct a thorough regression test before testing can continue. This section discusses what is required to develop a standard mechanism for evaluating and reporting these problems to the developers.

A test is considered as passing if the actual results meet the expected results. If unexpected issues arise, the severity of each problem is evaluated using a predetermined set of criteria. These criteria, referred to as a defect rating, can rank or prioritize a defect from most serious to least serious. The processor or test team usually establishes this defect rating classification before testing begins. To assist the test team in reviewing the defects encountered during the test, an example of a defect rating scale is listed in Table 5-2. This scale is prioritized from level 1 to level 5.

Table 5-2. Defect Rating Scale

Priority Level	Criteria
1	• Critical • Does not allow testing to continue • Major malfunctions in the system • Defect found in the processing component of the system
2	• Major component failure • Does not allow testing to continue • Defect or malfunction found in certain areas of the system • Problem must be resolved

Priority Level	Criteria
3	• Minor functional problem • Testing can continue • Functions in certain components do not work properly • Components can still work with other components of the system
4	• Minor issue • Testing can continue • Minor editing error found in a system component • Cosmetic change needed
5	• Minor issue • Testing can continue • Design clarification issue • Implementation issue • To be resolved by the state or processor • Future enhancements to the system

As a guideline, any defect with priority 1 or 2 is severe enough to require that all testing stop and be restarted after resolution of the discrepancy. Defects rated up to priority level 3 need to be regression tested after the problem is resolved. It is recommended that all level 4 priorities (up to approximately 20) be resolved and tested prior to placing the system in a production environment. All defects rated level 5, if not corrected during the test, can be addressed as a future enhancement.

After each problem has been evaluated and assigned a defect rating, the entire test team should give final approval. Once the results are assigned a defect rating, the problems will be indicated in the TPR. The TPR, exemplified in Appendix D, is a format used to list and sequentially number problems identified during the test. This systematic method of processing problems will assist the test team in prioritizing and resolving the problems.

Each TPR form is assigned a unique identification number and will vary in content depending on the state and their processor. The TPR should indicate whether the problem's status is open, closed, or pending. The main components of a TPR form include a full description of the identified problem and the steps required to resolve the issue.

Every TPR form should include the following:

- Defect/priority rating
- Date the TPR was created
- Date the problem was resolved
- Names and signatures of the individuals who administered the test
- Technical manager of the project
- Description of the problem or issue

- Description of the resolution

The TPR also tracks events associated with resolving the reported problem. Once a problem has been resolved or retested, the retest results can be placed in the same TPR report. The date the function was reassessed and resolved should be stated, along with an explanation on how the problem was resolved. Those who retested the system should be listed along with their signatures. TPRs help ensure that proper measures are taken to resolve a problem in a timely manner and to confirm that every system function is operating efficiently and as expected.

5.5 Testing Issues Log

Numerous activities and procedures are performed during a test, and often various activities are performed at once. To track what is occurring within the test environment, it is critical to record all test events and their results. At times the outcome of a test is not necessarily classified in the TPR as a problem. A testing issues log can track such test results. This section defines the testing issues log and identifies its advantages. An example of a testing issues log is included in Appendix H.

The testing issues log is a detailed record documenting all actions of the testers; it is an important tool that is used to record exactly what happened during the test. The issues log also serves as a method of gathering more information about the system. It assists testers when recording observations about the usability or ease of use of the system. For instance, notes taken in one section of the test might be useful to another test team member. Ideas or thoughts not mentioned in previous documentation can also be recorded. The log provides an efficient way of allowing each tester to document and share what was observed.

The testing issues log can vary in format from one test team to another. It can consist of the following fields to fully document all actions, results, and issues that transpire during a test:

- Date
- Unique reference number
- Test scenario reference number
- Name of tester
- Issue/comment/problem.

What is stated in a test plan and what is expected from executing a test script do not always occur as anticipated. Concerns or unexpected results can delay the test process or lead to the creation of additional test scripts or what-if scenarios. Observations made during testing must be documented and accounted for. The testing issues log tracks activities not stated in the test plan or test scripts and provides accurate documentation of what occurred during the test. Before a system goes live, it is important to resolve all issues documented in the testing issues log. Any further tests performed on the system functionality must be documented in additional testing issues logs.

5.6 EBT WIC REPORTS

After a test is conducted, all reports and outputs generated from the WIC system must be collected, and their contents verified. The layout, content, name, and even the reports produced, will vary from one WIC system to another. It is important to fully understand the contents of each output before interpreting its fields and validating the data contained in the fields. This section provides a list of standard reports generated by the WIC system as well as information concerning what must be done when interpreting these reports.

Reports can provide information pertinent to the system's users, administrators, developers, and clients of the WIC program. How and what the system generates can also influence managerial decisions. Therefore it is vital to validate the contents of the report after a thorough test of the system is complete. Every system's detailed design document provides a list of the processing system's standard reports and associated formats. All reports indicated in the design document must be analyzed, including all their data elements. The state must also understand how each report is produced and when the report is generated. Reports generated during testing and not mentioned in the design document should be included.

Every WIC system generates a group of reports daily and monthly. These reports validate the performance of all functionality of the system. Outputs generated from the WIC system can range from receipts produced when a client purchases a product at the grocer to a report listing benefits issued to a client's account. Examples of various WIC system-generated reports are listed below:

Daily Reports

- Audit Log
- Daily Transactions Report
- Daily WIC Account Report
- Benefit Authorization and Adjustments Report

Monthly Reports

- Monthly UPC Report
- Monthly WIC Accounting Report
- Monthly Manual Report

Settlement Reports

- FNS Drawdown Report

- System Balance Report

All reports generated by the WIC system must be validated. Checking the validity of the reports against the transactions and activities performed on a client's account ensures a system is working correctly. Numerous methods and procedures are used to verify the logic of each report and the information it displays. A few of these procedures are listed below:

- **Report Objectives:** For whatever purpose a report is generated, it is important the report accomplishes its objectives. Some reports are used to verify transactions performed on an individual's account, others track and list the number of invalid PIN attempts. The design document of a system should describe what each report intends to achieve.

- **Report Contents:** It is most important to verify that all required fields are displayed on every report. Missing, duplicate, or unmatched transactions should be verified on a report that provides transaction information.

- **Report Calculations:** If a purchase, return, or void is performed, the respective amounts and affected balances should be verified. If calculations are performed on a report, all totals and balances should be checked to ensure the amounts are displayed accurately.

6.0 TEST EXECUTION

Preparing for the test is necessary, but executing the test according to the test plan is yet more critical. Executing the test is the final step before the EBT system's pilot implementation. With proper planning, the execution of the test can be less stressful. Planning and organizing the test helps ensure the system functions are thoroughly evaluated according to the test scripts. This section will provide the necessary guidance to effectively execute test scripts and manage test activities.

Throughout execution of the test, certain activities and problems can be expected. Evaluation of the system should be performed according to the test plan, which details the approach and testing requirements. Test activities include environment overview, test coordination, script execution, problem management, result validation, documentation, and test reporting. These activities assist in managing test expectations.

6.1 Environment Overview

At the beginning of the test, it is important that the test environment be described to the participants. This exercise provides the testers with an understanding of the system components and functions to be evaluated. The test manager is responsible for presenting the system's description based on the design and functional requirements. The overview should also make the testers aware of functions or components that are not included or not working in the test environment. In addition, the overview for subsequent test(s) should address all issues identified from the previous testing effort. This helps to validate that all previously identified system problems were properly retested and resolved.

6.2 Test Coordination

Building on the guidance provided in the test plan, the test should be executed in an organized manner. After describing the test environment, test guidelines are made clear to the test participants by the test manager. The test guidelines will address test participation, script changes, problem logging, status meetings, what-if scripts, and TPR ranking or prioritizing.

Test Participation – This activity involves identifying an individual or team to execute a script or set of scripts for a particular functional area or for multiple functional areas. For redundancy in executing the test, assigning two or three testers to perform a set of scripts can help limit deviations from the scripts. The test scripts should be divided in to daily activities (e.g., Day 1, Day 2, Day 3). It is also a good idea to enlist a representative mix of test participants. If the participants include representatives from multiple organizations, each test team should include at least one representative from each organization. This allows for varying perspectives and promotes discussions on questions and issues. Establishing test teams in this manner also allows for independent validation of test scripts and their results.

Script Changes – The test manager is responsible for updating and maintaining the "master" scripts and must be made aware of all script changes or deviations when they occur. In some situations, a script performed incorrectly can be reversed or "backed out" allowing the tester to redo the script correctly.

Problem Logging – Problems and questions are expected to arise during testing. It is important that all perceived problems be documented on TPRs immediately. Prompt documentation of problems will help ensure that all the necessary details are captured. This is especially important when attempting to re-create a problem. If necessary, all information (receipts, screen prints, and reports) supporting the identified problem should be captured and attached to the specific TPR.

Status Meetings – A meeting of all test participants should take place at the beginning and end of each day. These meetings provide opportunities for group discussions concerning the problems identified. The morning meetings should be used to address the plan for the day's activities. This session can also serve as a follow-up to the meeting that occurred at the end of the previous day. Both daily sessions are helpful in providing problem details and background information that had not yet been addressed. In addition, the meetings enable the ranking of all identified problems by the team participants.

What-if Scripts – What-if scripts are documented on a form; they test additional system functions or areas that were not address in the master scripts. These scripts, in combination with the master scripts, allow test participants to conduct a more robust evaluation of the system. Any test participant can create a what-if script at any time before or during the test. However it is important that the test manager and the testers (if necessary) review the proposed scripts at least one day before the test is conducted. This review helps prevent duplication of scripts already addressed in the master scripts. The review also helps the test manager determine whether sufficient time and resources are available to set up and execute the scripts. Some scripts may require certain actions to be performed before the script can be completed. For example, a card replacement in a smart card-based WIC EBT system requires at least two days to transfer benefits from the old card to the new. It is therefore highly recommended that all possible what-if scripts be identified as early as the first or second day of testing.

TPR Ranking – All problems identified during the test should be documented on TPRs (see Appendix D). Each problem should also be discussed at the end of each test day and ranked by the test participants. A discussion concerning the status of all TPRs should take place every morning following a day of testing.

6.3 Script Execution

Execution of a test script is considered successful only when all steps have been performed and the expected results match the actual results. The test scripts should be performed exactly as written in the master script. Supporting information, such as screen printouts, receipts, and notes, should be attached to each "master" script, TPR, or what-if script. When executing test scripts, any perceived problem concerning the master scripts or test results should be discussed with the test manager.

In some cases the master script can be written incorrectly, resulting in disparate expected and actual results. It is important that all problems or deviations associated with the master scripts be identified, documented, and resolved. Settlement results of the test can be forecasted if the test is well planned and organized, whereas undocumented changes or deviations can affect the expected outcome. Overlooked or incorrectly entered transactions will change the settlement results.

6.4 Problem Management

All problems identified during the test must be documented on a TPR and prioritized or ranked (see Section 5.4). The TPRs (see Appendix D) should indicate information such as functional area, TPR number, problem description, and corrective actions. TPR information is important to the test process because it provides a method for distinguishing a major issue from a minor or simple one. Ranking is especially helpful when trying to determine the severity of a problem. All major problems should be corrected before pilot implementation.

6.5 Results Validation

Preparing for the test, executing the test according to the scripts, and recording the results are critical aspects of testing. However, without validating test results via daily reports, the accuracy of the system's settlement reporting process is unknown. Results validation entails reconciling printed test results against system-generated daily activity reports.

Test participants are responsible for ensuring that test activities are validated. Test documentation at the conclusion of a test will be instrumental in validating the results. The documentation should provide sufficient information to allow test participants to re-create any given test scenario. Test activity performed using components such as the WIC clinic system and POS devices can be validated using screen printouts, POS receipts, master scripts, and settlement reports. Screen printouts and the master scripts should be used to validate WIC clinic activities. POS receipts and the master scripts should be used to validate POS transactions that were performed. These receipts and scripts can also be used to validate the information provided on the settlement reports. Other test activities can also be validated using screen printouts, POS receipts, master scripts, settlement reports, and other ad hoc reports.

6.6 Test Reporting

Test reporting captures the activities and problems identified and discussed during the test. After completion of the test, test reporting should also include any new problems that were not identified during the test but discovered from further test documentation. The importance of an organized process to keep account of test problems will be realized during test reporting. Key areas detailed in the report include the following:

Functional Areas – The report should discuss functional areas that were evaluated, as well as areas excluded from the test and the reasons for their exclusion.

Test Environment – The configuration of the environment that was evaluated should be illustrated in the report. In addition the report should list and discuss the key system components, reflecting the relationships between WIC EBT functions and components.

Test Problems – Test problems should be discussed according to their functional areas. The status (e.g., open, closed) of all identified problems should be included in the discussions. All supporting information concerning a particular problem should also be provided. If a problem was resolved, the report should state how. Providing a full account of all problems will assist the reader in understanding why the problems occurred and how they were resolved.

Other areas that can be addressed in the report include the organization(s) that participated in the test and a status of outstanding issues from prior testing. The report may even provide considerations and recommendations concerning the closure of TPRs or functional enhancements for the system.

6.7 Report Review

During the acceptance test, the tester should be able to produce printouts of all reports, including the daily activity report, settlement reports, and all routine reports the system is expected to produce when normal operations commence. Each report should be carefully reviewed while the system is still in the test phase. Once the system goes live and is in production, it may be difficult to detect or correct problems that surface in the reports. It is therefore critical to perform daily review of reports to ensure that each report balances, reflects the previous day's activities, and that the data represented is correct.

Reports are crucial in enabling the tester to detect errors before the system goes into production. It is also important to examine the reports from the perspective that sufficient data is available to monitor the system during production. For instance, a user must understand how daily activities are represented in reports. And if a problem is reported, it’s equally critical to know how to use reports to troubleshoot the problem.

7.0 SUMMARY

The principles, methodologies, objectives, and myths identified in this document have been provided to assist state agencies as they prepare for and ultimately execute a WIC EBT system test. System testing is potentially the most important aspect of developing a WIC EBT system. This document also provides the tools and guidance for planning, conducting, documenting, and executing a successful system test.

Because it is difficult to predict the time required for test completion, the test plan is a significant test tool. System knowledge is yet another important tool; states will have unique WIC EBT features that need to be understood to achieve successful testing and implementation. The following "best practices" will assist states as they test their WIC EBT systems:

- Develop a well-organized test plan, detailing the test activities to occur each day.
- Provide an overview of the system at the beginning of the first day of testing.
- Two daily meetings (morning and afternoon) should occur to address status, problems, questions, and next steps.
- The test manager should coordinate and discuss the test scripts and activities with test participants.
- The test manager should manage the test execution process to quickly identify and resolve problems and script deviations.
- Test participants should log all problems in TPRs.
- All supporting (test) problem information should be attached to the TPR.
- All problems should be discussed, ranked, and approved during a group discussion with the test participants.
- All test activities should be validated.
- Determine in advance the individuals involved in the go/no-go committee.
- All errors, including those that are low-priority, should be resolved before proceeding to pilot implementation.
- Test participants should include only those parties listed in Table 4-1. An audience of curious spectators or other potential contractor clients could add pressure to rush the test.
- Data used for testing should resemble real-life data, such as "1 quart of milk" instead of "1 liter of milk."

By thoroughly preparing for a test and conscientiously executing a comprehensive set of well-defined test scripts, test participants will be able to detect the majority of defects in system software. Once the defects are identified, they must then be prioritized and resolved accordingly. Through the use of tools such as an RTM, test participants will not only have a roadmap to guide them through the system, but they will also have an excellent tool to help them ensure all user-defined requirements have been addressed.

Appendix A – Glossary of Terms

Child Nutrition Act of 1966 – An Act of Congress that established numerous food assistance and education programs designed to provide assistance to lower-income families and at-risk children. These programs include the WIC, School Lunch, and School Milk programs.

Detailed Design Document (DDD) – A document that serves as the foundation of an application development effort. It is the developer's interpretation of the requirements as defined by the users. After the document is finalized and signed off by the users, the developers may begin building the various components of the system.

Electronic Benefits Transfer (EBT) – The conversion of paper-based benefit delivery instruments (vouchers or checks) to electronic payment applications, which ensure that the commercial networks (point of sale [POS] networks) acquire transactions and that the commercial banking system settles those transactions.

Magnetic Stripe Card (Mag Stripe) – Plastic card technology designed to electronically retain basic information, including the primary account number (PAN) and personal identification number (PIN) offset. Transactions must be executed at terminals (such as POS devices) that have on-line access to authorizing systems.

Production – After an application has been deemed ready to receive and generate real transactions, it is released into this environment.

Requirements Traceability Matrix (RTM) – A tool that lists all user-defined requirements in table form. The matrix can be used as a checklist to ensure that all requirements have been built into the system, and as a roadmap for testers to follow.

Smart Card – A plastic card embedded with a computer chip used to store relevant cardholder information, such as PAN, PIN, and WIC benefits information.

Special Supplemental Nutrition Program for Women, Infants, and Children (WIC) – A program for providing food and nutritional education to needy Women, Infants, and Children. The program is administered by the states and local agencies and is funded by the Federal Government.

System Requirements Specification (SRS) – A document that describes every function, feature, and interface (internal and external) that must be built into a system. The document is sometimes referred to as a Requirements Document. Construction of the document is typically a team effort that includes the following participants: management, system users, system designers/developers, testers, system trainers, and customer service representatives. System designers and develops use the document to generate the DDD.

Test Coordinator – The individual responsible for planning, managing, and documenting a test effort. This person may or may not actively participate in the testing but has overall responsibility for conducting and finalizing the test.

Test Manager – A more formal title for Test Coordinator.

Universal Product Code (UPC) – A bar code that is printed on a package to precisely identify its contents. The codes are used with bar code-scanning cash registers for pricing and can also be used as part of an automated inventory management process.

Appendix B – Acronyms and Abbreviations

ACH	Automated Clearing House
CFR	Code of Federal Regulations
DDD	Detailed Design Document
EBT	Electronic Benefits Transfer
FNS	Food and Nutrition Service
ID	Identification
NSF	Non-Sufficient Funds
PAN	Primary Account Number
PIN	Personal Identification Number
PL	Public Law
POS	Point of Sale
RTM	Requirements Traceability Matrix
SRS	Software Requirements Specification
TPR	Test Problem Report
UAT	User Acceptance Test
UPC	Universal Product Code
USDA	United States Department of Agriculture
WIC	Special Supplemental Nutrition Program for Women, Infants, and Children

Appendix C – Testing Checklist

The following checklist indicates system setup items that should be prepared before a test:

- Most recent software version loaded
- System parameters set correctly
- Test data and hardware configuration consistent with test plan
- All necessary communication links available

The following checklist indicates practices that will assist states in testing WIC EBT systems:

- Develop a well-organized test plan, detailing the test activities to occur each day.
- Provide an overview of the system at the beginning of the first day of testing.
- Two daily meetings (morning and afternoon) should occur to address status, problems, questions, and next steps.
- The test manager should coordinate and discuss the test scripts and activities with the test participants.
- The test manager should manage the test execution process to quickly identify and resolve problems and script deviations.
- Test participants should log all problems in TPRs.
- All information supporting test problem should be attached to the TPR.
- All problems should be ranked with group discussion and approval.
- All test activities should be validated.

Appendix D – Sample TPR Form

TPR No. ________
Priority No.________

TEST PROBLEM REPORT

Status __________

Tester Name ________________________________
Test Component ____________________________

Date __________
Version No.________

Problem Description
Card Number ________

____
____
____
____
____
____

Problem Resolution

____
____
____
____
____
____

Owner ____________________________ **Date** __________________

Appendix E – Requirements Traceability Matrix (RTM)

Requirements Reference Number	System Requirement	Component	Detailed Design Document Reference	Test Script Reference	Test Verification Pass / Fail	
Card Replacement						
UR001	Client may call the ARU number to report card as lost/stolen/damaged.	ARU, help desk, clinic	Section 4.0	34		
UR002	System shall status card as being lost/damaged/stolen.	Host system, help desk, clinic	Section 4.2	34		
UR003	Reissue card resulting from lost/stolen/damaged.	Host system	Section 4.3	36		
UR004	A new PAN will be assigned to a reissued card.	Host system	Section 4.3	38		
UR005	The system will produce a new card.	Host system	Section 4.3	38		

Appendix F – Sample Test Script Template

Benefit Package Information

Food Package ID	Description	Category	Sub-Category	Food Package	Quantity
0055	Pregnant Mothers	01	000	Milk	8 gallons
		06	001	Beans	5 cans
		02	000	Cheese	5 lbs
		03	000	Eggs	1 dozen
0056	Infant	09	000	Infant cereal	4 boxes
		10	005	Infant juice	64 oz
		11	000	Formula	6 cans
0057	Child	01	001	Milk	2 gallons
		06	002	Peanut Butter	16 oz
		02	000	Cheese	3 lbs
		05	000	Cereal	24 oz

WIC UPC Table

Item #	Product Name	Category	Sub-Category	UPC	Description/ Unit of measurement	Price
1	Milk	01	000	000552898	1 gallon whole milk	$3.25
2	Cheese	02	000	400021433	1 pound	$1.25
3	Eggs	03	000	000234574	1 dozen – medium eggs	$1.20
4	Cereal	05	000	256645302	32 oz	$2.75
5	Beans	06	001	980043661	1 can	$.75
6	Peanut Butter	06	002	432140074	16 oz	$2.99
7	Infant Cereal	09	000	732282100	8 oz	$1.35
8	Infant Juice	10	005	565550070	8 oz	$1.00
9	Infant Formula	11	000	000111430	1 can	$2.50

Case Information

Type	Value
Case Number	1
Cardholder Name	Betty Bop
PAN Number	1000000000011111
Card Status	Active
Food Package ID	0055

Betty Bop – 1000000000011111

Transaction Number	Transaction	Steps	Beginning Balance	Ending Balance	Expected Results	Actual Results
1	**Add Infant to Case**	1. Insert card at the clinic 2. Locate case on the clinic screen 3. Add new member to family account 4. Input infant information 5. Write new data to update card	▪ 8 gallons milk ▪ 5 cans beans ▪ 5 lbs cheese ▪ 1 dz eggs	▪ 8 gallons milk ▪ 5 cans beans ▪ 5 lbs cheese ▪ 1 dz eggs	Infant added to account	
2	**Add infant's food benefits to card**	(Clinic representative is still logged onto the same family account) 1. Go to Add Benefits screen 2. Find Benefit Package for Infants 3. Issue package to case 4. Write to card	▪ 8 gallons milk ▪ 5 cans beans ▪ 5 lbs cheese ▪ 1 dz eggs	▪ 8 gallons milk ▪ 5 cans beans ▪ 5 lbs cheese ▪ 1 dz eggs ▪ 4 bxs infant cereal ▪ 64 oz infant juice ▪ 6 cans formula	Benefits updated to card	
3	**Change PIN**	1. Select change PIN from menu 2. Client enters 4-digit old PIN 3. Client enters new 4-digit PIN 4. Client reenters new 4-digit PIN	N/A	N/A	PIN changed	

Case Information

Type	Value
Case Number	2
Cardholder Name	Dee Jones
PAN Number	1000000000022222
Card Status	Active
Food Package ID	0056

Dee Jones – 1000000000022222

Transaction Number	Transaction	Steps	Beginning Balance	Ending Balance	Expected Results	Actual Results
1	**Purchase WIC Items**	1. Swipe card 2. Enter valid PIN 3. Scan 2 boxes cereal 4. Scan 16 oz baby juice 5. Update account 6. Remove card	▪ 4 boxes cereal ▪ 64 oz infant juice ▪ 6 cans formula	▪ 2 boxes cereal ▪ 48 oz infant juice ▪ 6 cans formula	3. Approved 4. Approved	
2	**Purchase WIC and Non-WIC Items**	1. Swipe card 2. Enter valid PIN 3. Scan 1 carton of cigarettes 4. Scan 2 cans formula 5. Scan 1 lb bacon 6. Update account 7. Remove card	▪ 2 boxes cereal ▪ 48 oz infant juice ▪ 6 cans formula	▪ 2 boxes cereal ▪ 48 oz infant juice ▪ 4 cans formula	3. Denied 4. Approved 5. Denied	
3	**Return**	1. Swipe card 2. Enter valid PIN 3. Scan 2 g milk 4. Scan 2 16 oz infant juice 5. Void last – 1 16 oz baby juice 6. Update account 7. Remove card	▪ 2 boxes cereal ▪ 48 oz infant juice ▪ 4 cans formula	▪ 2 boxes cereal ▪ 32 oz infant juice ▪ 4 cans formula	3. Denied 4. Approved 5. Void 6. Approved	

Case Information

Type	Value
Case Number	3
Cardholder Name	Jane Doe
PAN Number	1000000000044444
Card Status	Active
Food Package ID	0057

Jane Doe - 1000000000044444

Transaction Number	Transaction	Steps	Beginning Balance	Ending Balance	Expected Results	Actual Results
1	**Lock Card** – Invalid PIN attempts	1. Swipe Card 2. Enter invalid PIN (1st try) 3. Enter invalid PIN (2nd try) 4. Enter invalid PIN (3rd try) 5. Enter invalid (4th try) 6. Terminal will display “Resolve at the WIC clinic”	N/A	N/A	2. Denied 3. Denied 4. Denied 5. Denied – card locked	
2	**Attempt WIC items purchase**	1. Swipe card 2. Enter valid PIN			Denied – card locked	
Card Resets 12:00 am next day						
3	**Purchase more than available on account**	1. Swipe card 2. Enter valid PIN 3. Scan 1 g milk 4. Scan 8 oz peanut butter 5. Scan 4 lb cheese 6. Update account 7. Remove card	▪ 2 g milk ▪ 16 oz peanut butter ▪ 3 lbs cheese ▪ 24 oz cereal	▪ 1 g milk ▪ 8 oz peanut butter ▪ 24 oz cereal	3. Approved 4. Approved 5. 3 lbs cheese – approved 1 lb cheese – Denied	
4	**Report Card Lost**	1. Call ARU 2. Report card lost 3. Card placed on the Hot Card List			Card statused	

Transaction Number	Transaction	Steps	Beginning Balance	Ending Balance	Expected Results	Actual Results
5	**Attempt to Purchase WIC items**	1. Insert card into the POS terminal 2. Card denied; terminal will display “Resolve at WIC Clinic”			Card denied at POS terminal	

Appendix G – Sample What-if Testing Scenario Form

Test Scenario Number	What-if Scenario	Expected Results	Actual Results	Comments
1.				
2.				
3.				
4.				
5.				

Appendix H – Testing Issues Log

Date	Issue Number	Functional Area	Component	Description	Resolution	Tester
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	10					

Appendix I – References

1. Stored Value Systems (SVS), Wyoming, *EBT System Rollout Acceptance Test Plan*, *Final Version 1.6*, August 18, 1999.
2. Stored Value Systems, Ohio, *WIC Pilot System Multi-Program Chip Card Food Stamps and WIC Acceptance Test Plan*, August 7, 2000.
3. American Bankers Associations, District of Columbia, *X9.93 2002, Financial Transaction Messages – Electronic Benefits Transfer (EBT), Part 1: Messages, Draft version 5.1,* 2002.
4. American Bankers Associations, District of Columbia, *X9.93 2002, Financial Transaction Messages – Electronic Benefits Transfer (EBT), Part 2: Files, Version 5.3,* 2002.